Microsoft

.net developer training

Create XML Web Services and Applications with Visual Studio .NET

The .NET Framework

Module 02 Agenda

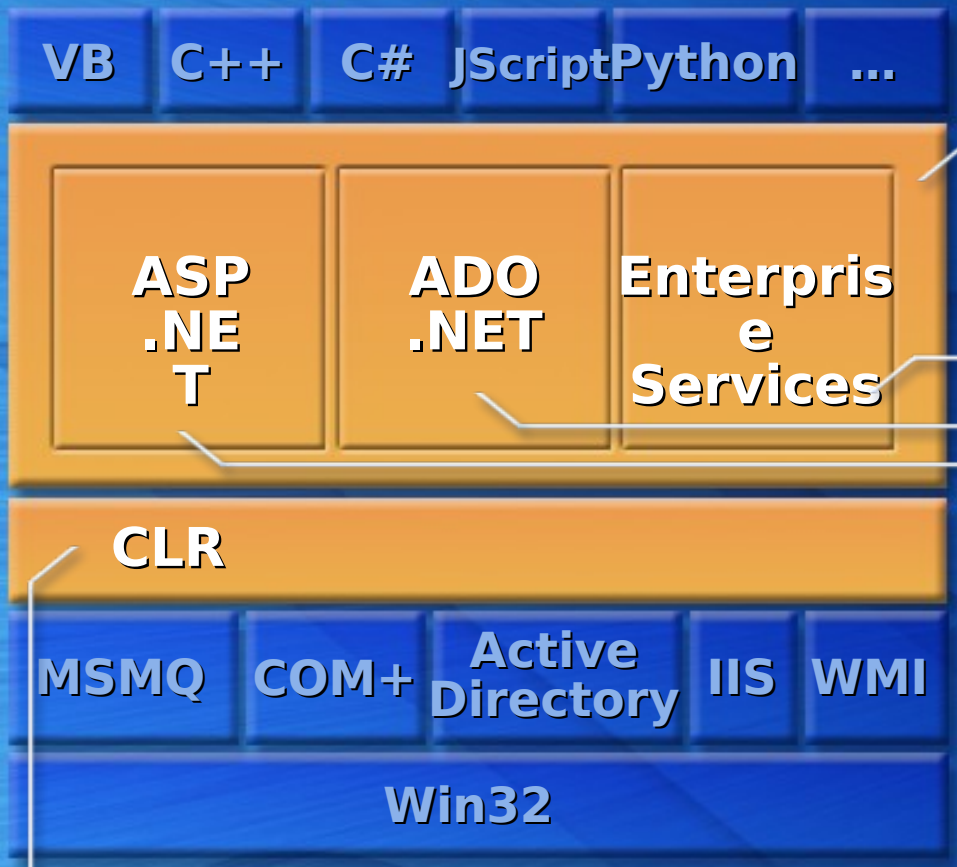
- Inside the .NET Framework
- Common Language Runtime, CLR
- The .NET Unified Programming Model and Class Library
- Multi-Language Development
- Simplified Deployment
- Integration
- Simple to Use

chass:

the .NET Framework
Infrastructure for the .NET Platform

- Common Language Runtime
  - Manages running code
  - Verifies type safety
  - Provides garbage collection, error handling
  - Security
  - Provides common type system
  - Provides access to system resources
    - Native API, COM interop, etc.
- A collection of unified classes
  - ASP.NET
  - ADO.NET
  - Enterprise Services
  - System, System.Drawing, System.Web, System.XML, etc.

Win32 and the .NET Framework



Unified Class libraries

Unifies Programming models across languages

Factored for extensibility

Designed for tools

Enterprise Services

The services required for mission critical applications: Transactions, Messaging, Partitions, Object Pooling and Events

XML and Data Access

ADO .NET interfaces any database, loosely coupled data access and native XML data format; leverages the huge library of ODBC OLE DB drivers

XML Web Services and Scripting

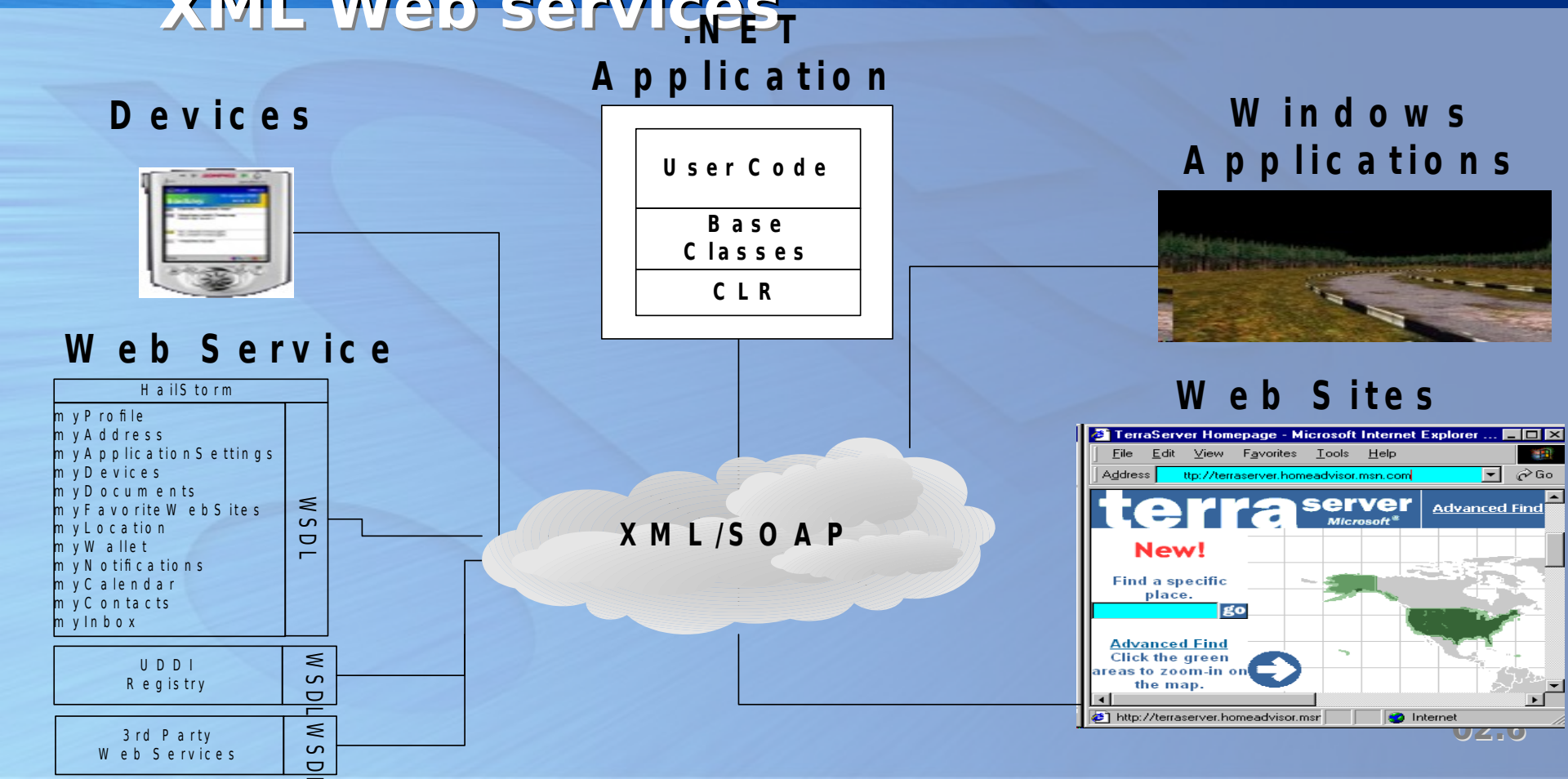
High-productivity environment for building and operating XML Web services

CLR

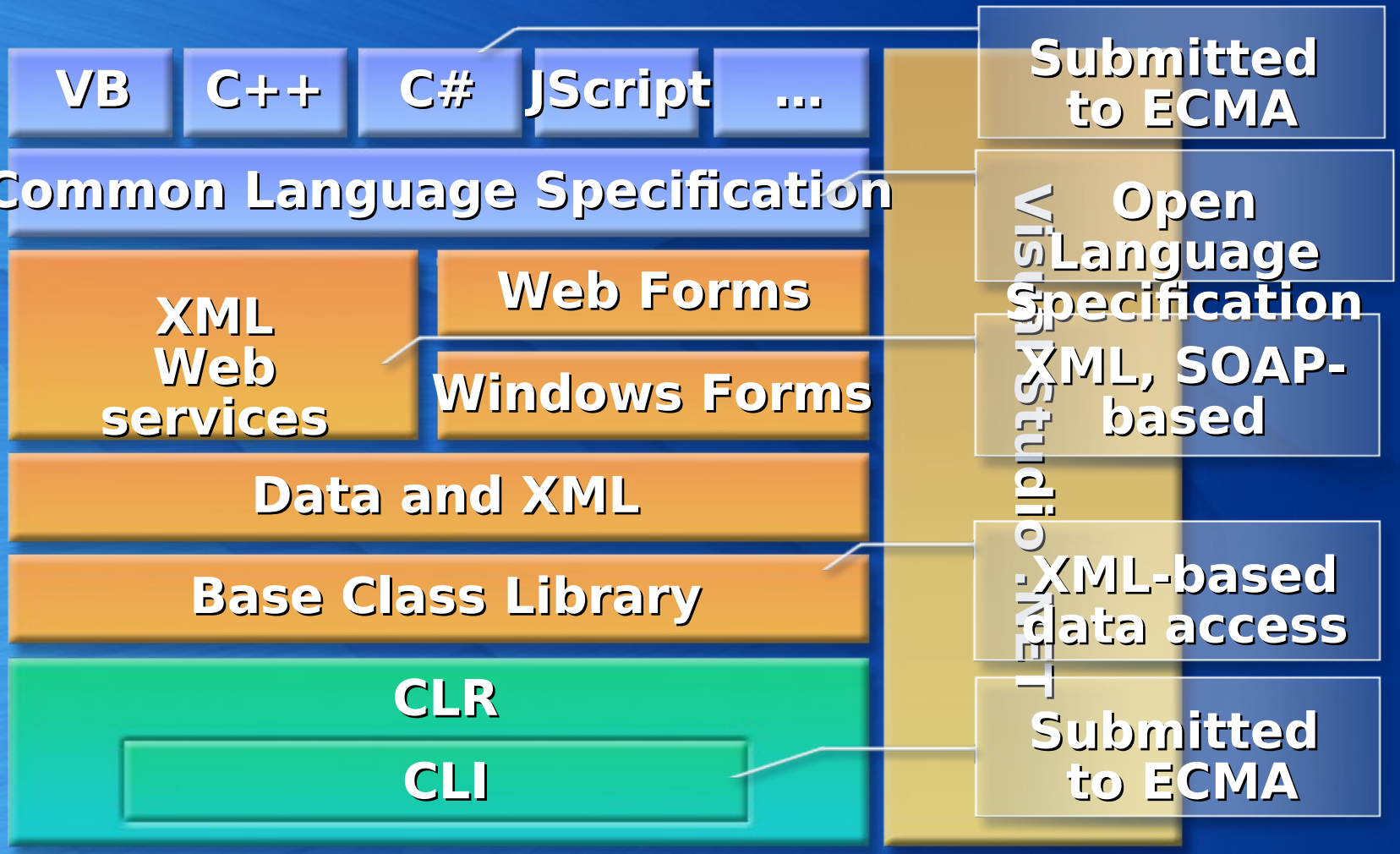
Executes code, maintains security, handles component “plumbing” and dependencies

Inside the .NET Framework
Web Services, Windows Applications and more

- .NET Framework favors loosely connected, distributed and stateless XML Web services

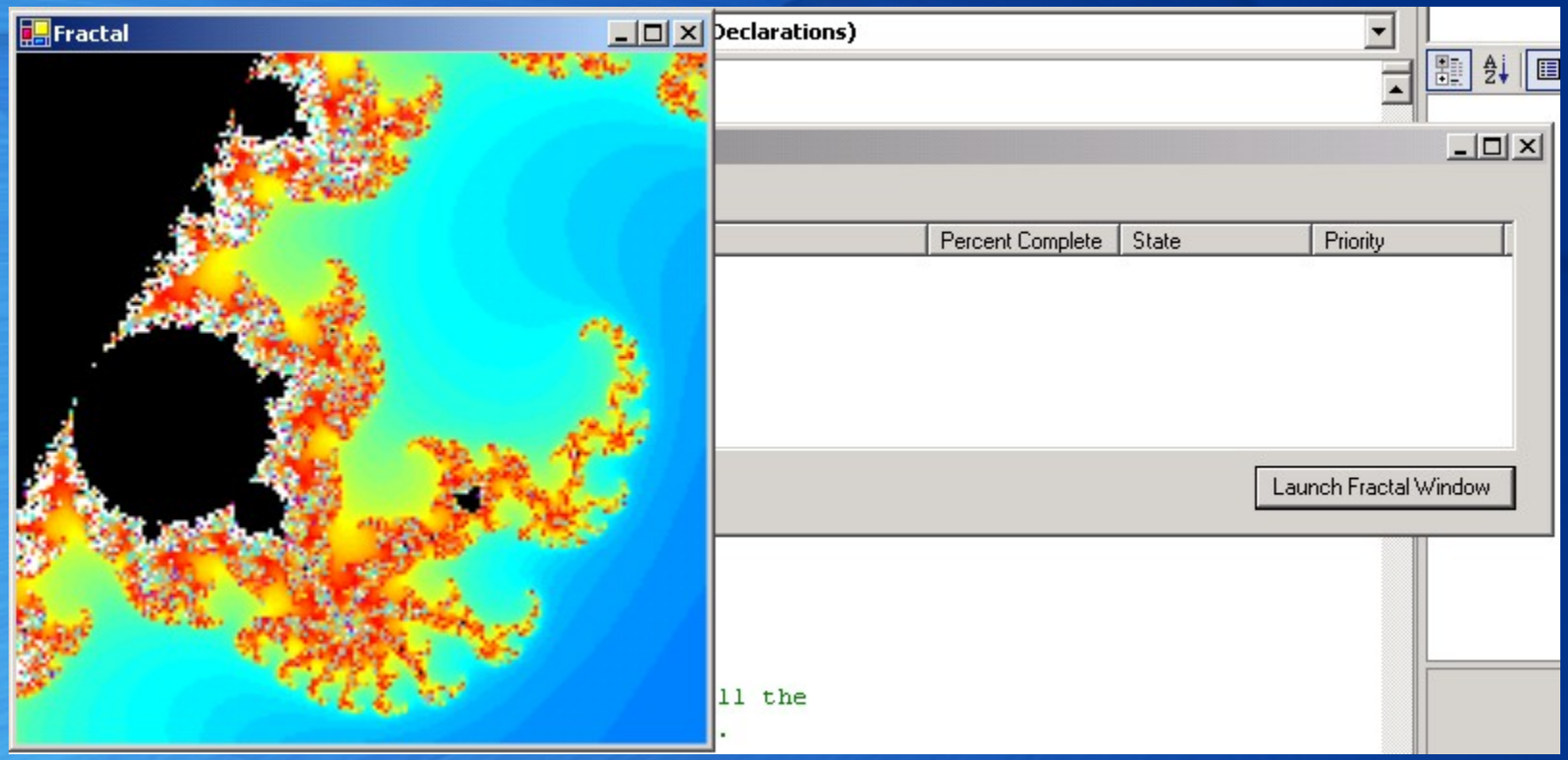


Inside the .NET Framework
Based on Standards



The .NET Framework
Demo

- Fractals and Threading
- Demonstrate running fractal sample



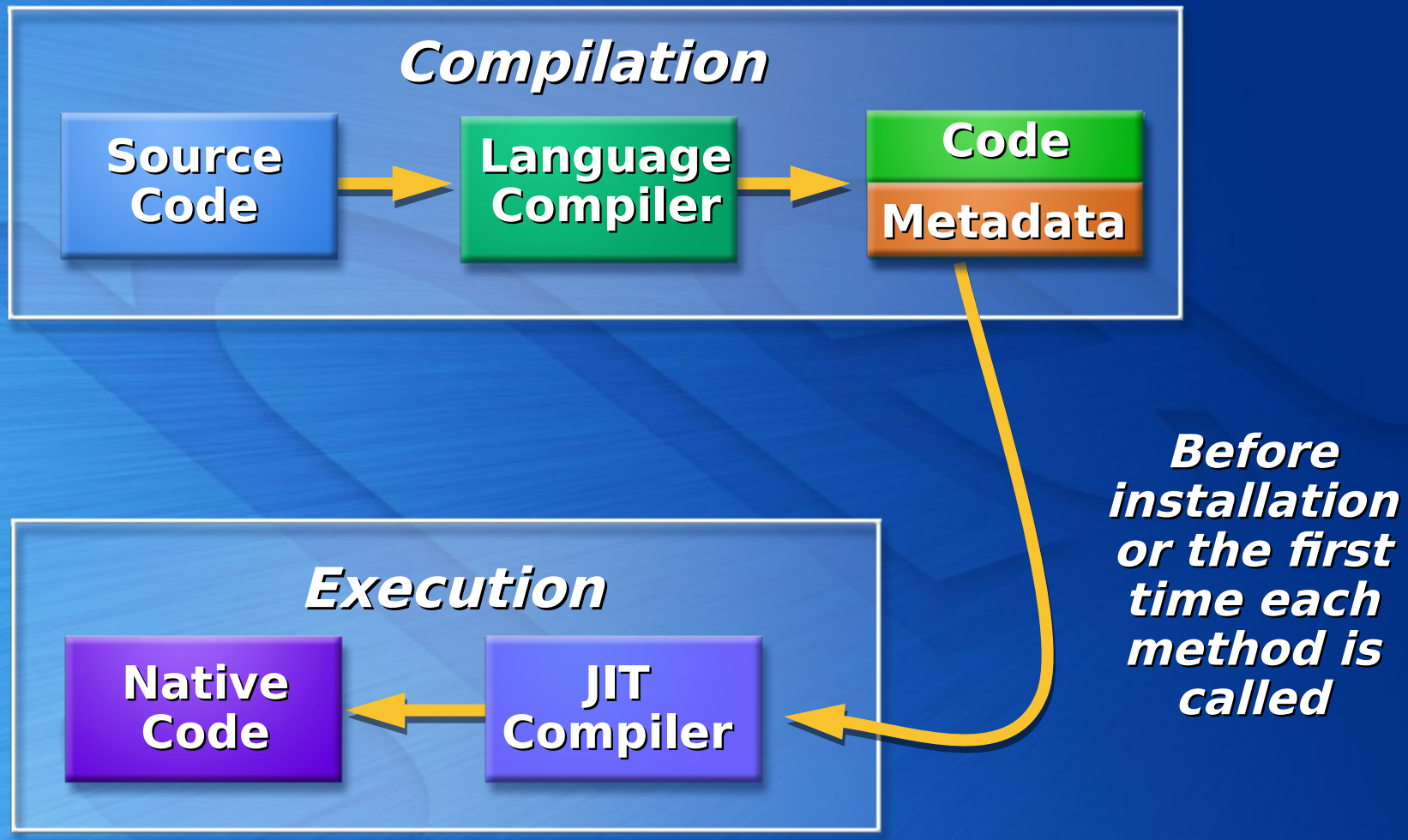
chass:

on Language Runtime

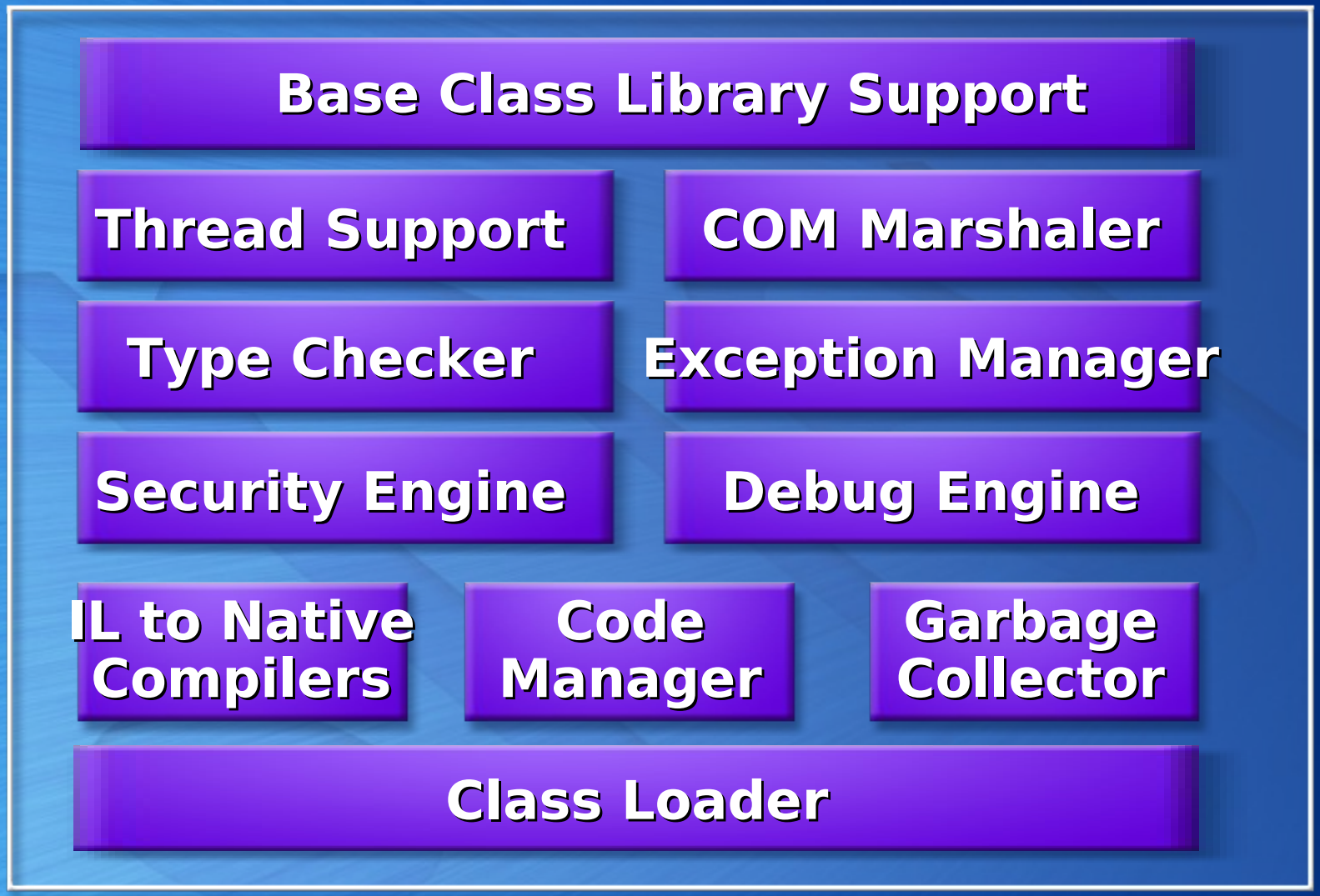
Features

- Manages running code
  - Threading
  - Memory management
  - No interpreter: Install time or run time IL to native compilation
- Fine-grained evidence-based security
  - Code access security
    - IL can be verified to guarantee type safety
    - No unsafe casts, no uninitialized variables and no out-of-bounds array indexing
  - Role-based security
  - Integrated with underlying OS
- XCOPY deployment
- Object remoting with SOAP

CLR
Compilation and Execution



CLR
Internals



CLR
Simplified Development

- Eliminates COM plumbing
- No more …
  - Registration
  - GUIDs
  - .IDL files
  - HRESULTs
  - IUnknown
  - AddRef/Release
  - CoCreateInstance

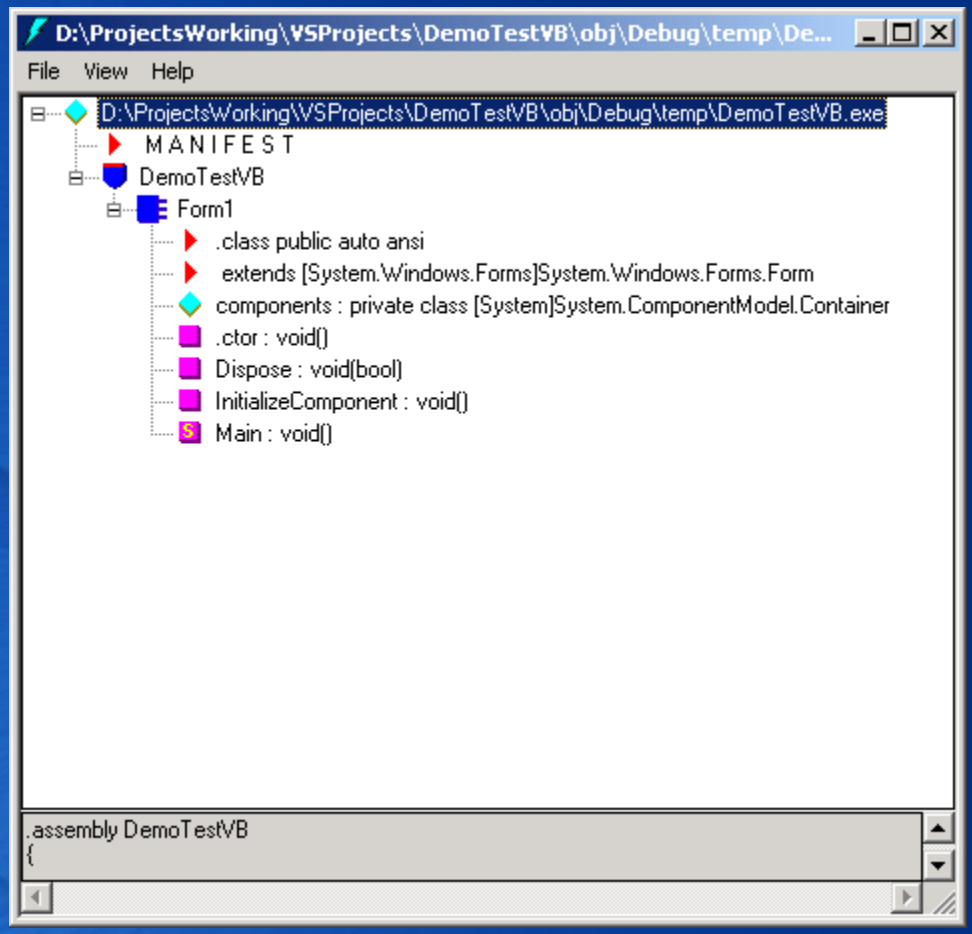
=>self describing components
=>hierarchical namespaces
=>source code to metadata
=>structured exceptions
=>root object class

=>garbage collector

=>”new” operator

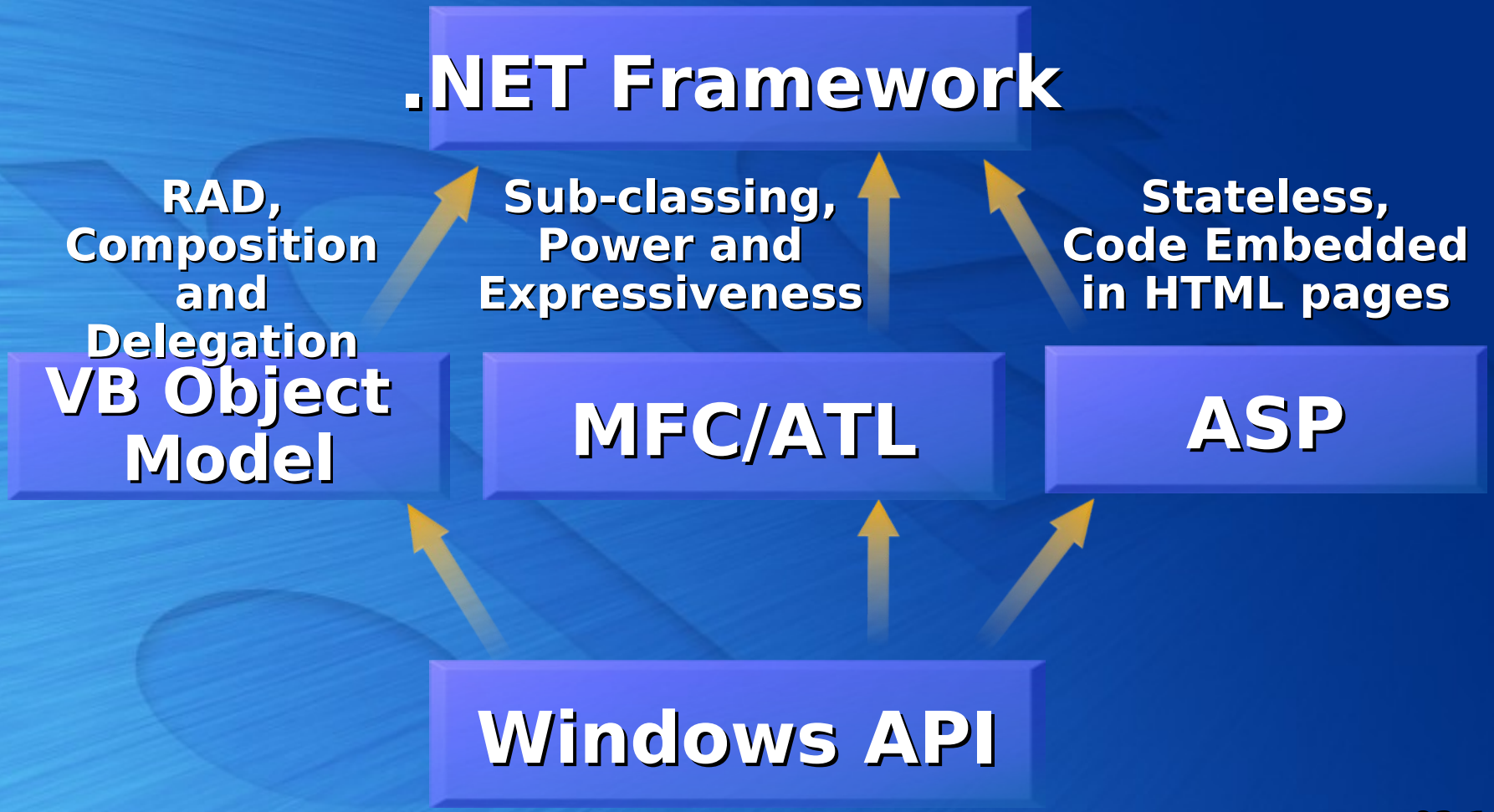
CLR
Demo

- CLR Basics Using ILDasm

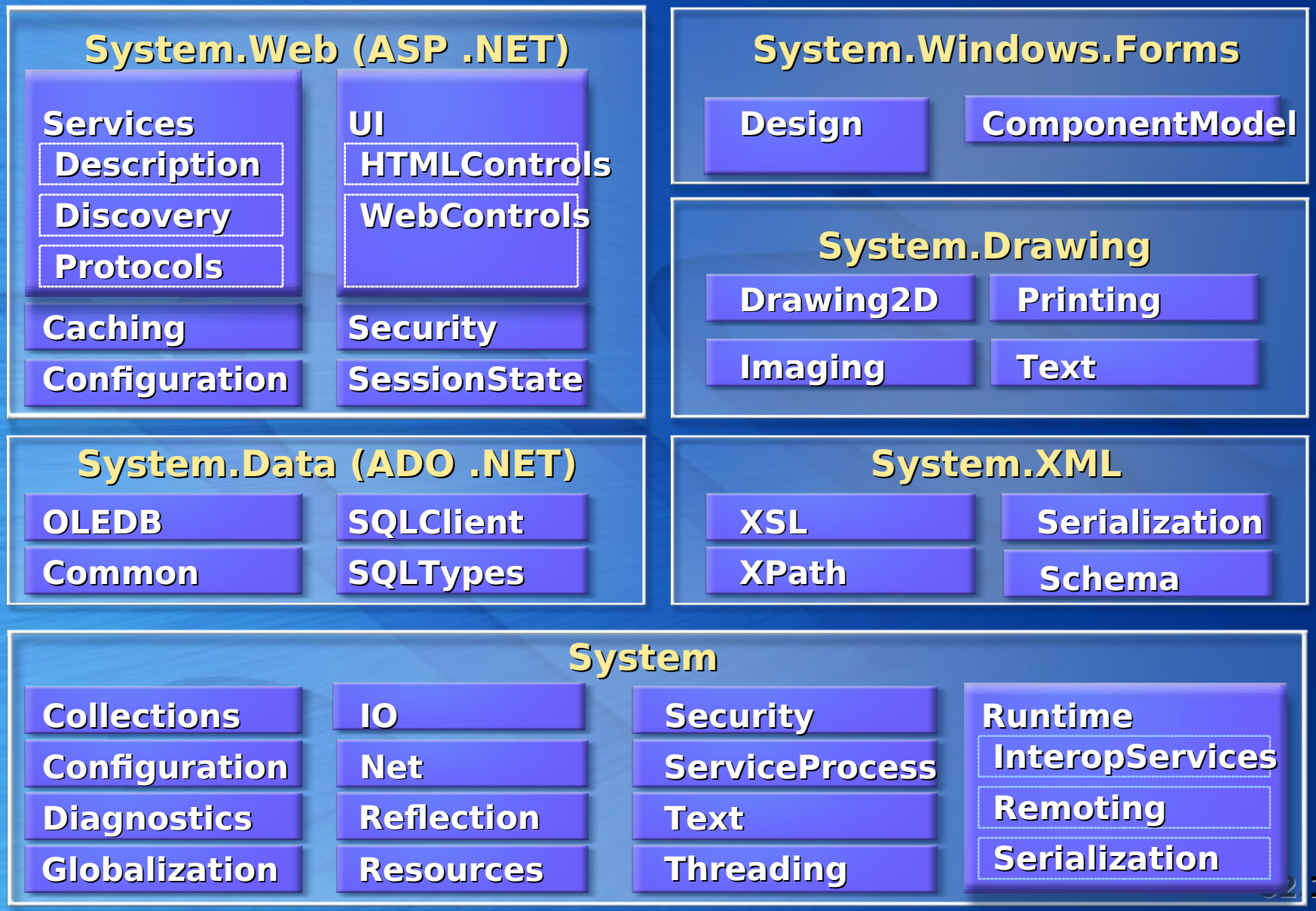


.NET Unified Programming Model

Consistent API availability regardless of language and programming model



.NET Unified Classes



Multi-Language Development

- Object system is built in, not bolted on
  - Use the language of choice
  - No additional rules or API to learn
- Advanced multi-language features
  - Cross-language inheritance and exceptions
- Supports more than 20 languages
- Tools that work in multiple languages and across languages and platforms
  - Debugger

Simplified Deployment

- **No registration required**
  - Code is self-describing
  - Copy components to application directory
- **Zero impact install**
  - Installing one application fails to affect another
- **Side-by-side execution**
  - Multiple component versions can co-exist

Seamless Integration

- Any .NET class can be a COM object
- COM objects can be imported
  - No ref counting or COM API necessary
- .NET classes use COM+ services
  - Transactions, Object pooling, etc…
- Easily call system functions
  - No marshaling for common cases
- Use XML Web services from Web or Windows applications

Simple to Use

- Component-oriented
  - Properties, methods and events are first class
  - Design-time functionality
- Organization
  - Organized in hierarchical namespaces
- Unified Type System
  - Everything is an object
  - Boxing converts value types to objects
    - No need for special variant type

Summary

- Common framework for all languages
  - CLR
  - Unified classes
- .NET is based on standards
- Simplified development and deployment
  - Multi-language support
  - Works with COM
  - Assemblies

Resources

- ***Frequently Asked Questions about Microsoft® .NET Framework -*** **http://msdn.microsoft.com/library/techart/faq111700.htm**
- ***The GotDotNet Home Page -*** **An excellent site to find more information about .NET with samples and real-world examples of .NET applications http://www.gotdotnet.com/**
- ***An Introduction to the .NET Framework -*** **A concise introduction to the .NET Framework http://www.gotdotnet.com/about_new.aspx**

In-Depth Training On The .NET Framework

Direct from Microsoft

- 2349A: Programming the Microsoft .NET Framework with C# (Prerelease)
- 2350A: Securing and Deploying .NET Assemblies (Prerelease)
- 2415A: Programming the Microsoft .NET Framework with Visual Basic .NET (Prerelease)
  - Available September 2001

Microsoft®

02.24